



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/671,215	09/25/2003	Bruce R. Booth	AUS920030296US1	4076
35525 7590 11/17/2008				
IBM CORP (YA)				
C/O YEE & ASSOCIATES PC				
P.O. BOX 802333				
DALLAS, TX 75380				
EXAMINER				
FEARER, MARK D				
ART UNIT		PAPER NUMBER		
2443				
NOTIFICATION DATE		DELIVERY MODE		
11/17/2008		ELECTRONIC		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ptonotifs@yeciipaw.com

# Office Action Summary

**Application No.**

10/671,215

**Applicant(s)**

BOOTH ET AL.

**Examiner**

MARK D. FEARER

**Art Unit**

2443

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 13 August 2008.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-5 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-5 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SF/ICE)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

### DETAILED ACTION

1. Applicant's Amendment filed 13 August 2008 is acknowledged.
2. Claim 1 has been amended.
3. Claims 6-27 are cancelled.
4. Claims 1-5 are still pending in the present application.
5. This action is made FINAL.

### *Claim Rejections - 35 USC § 103*

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various

claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

7. Claims 1 and 4-5 are rejected under 35 U.S.C. 103(a) as being unpatentable over Helgeson et al. (US 7072934 B2) in view of Patiejunas (US 20050108710 A1).

Consider claim 1. Helgeson et al. discloses a method for a business application server management system platform providing a single automation tool adapter within said business application server management system platform for use with multiple different automation tools, said method comprising the steps of: each one of said automation tools being an application that interfaces with a device to enact a sequence of automated changes (Helgeson et al., column 7 line 34 – column 8 line 11); said automation tool adapter including a queue listener; monitoring, by said queue listener, a plurality of clients by monitoring a request queue that is external to said automation tool adapter for messages (Helgeson et al., column 3 lines 18-45); determining, by said queue listener, whether said request queue includes an automation message (Helgeson et al., column 4 lines 29-51); said automation message including a request having parameters in an original format that are to be executed using one of said automation tools, said automation message including a message identifier that uniquely identifies said automation message and a source of said automation message (Helgeson et al.,

column 15 line 9 – column 16 line 64); responsive to said queue listener determining that said request queue includes said automation message, said queue listener creating a verb dispatcher within said automation tool adapter; passing, by said queue listener, said automation message to said verb dispatcher; creating, by said verb dispatcher, an automation processor of a type that is required by said particular verb (Helgeson et al., column 37 lines 42-46); translating, by said verb dispatcher, said parameters from said original format to a second format that is required by said one of said automation tools to produce translated parameters (Helgeson et al., column 119 line 28 – column 120 line 22); providing, by said verb dispatcher, said message including said message identifier and said translated parameters to said automation processor (Helgeson et al., column 4 lines 6-28); constructing, by said automation processor, an automation tool command using said translated parameters; executing, by said one of said automation tools, said automation tool command, said automation tool command including said translated parameters (Helgeson et al., column 14 lines 35-54); in response to a completion of execution of said automation tool command, said one of said automation tools sending a reply to a reply server that is included in said automation tool adapter, said reply being in said second format and including said message identifier; translating, by said reply server, said reply into said original format to form a translated reply (Helgeson et al., column 113 lines 1-33); and sending, by said reply server, said translated reply to a reply queue that is external to said automation tool adapter (Helgeson et al., column 114 lines 11-25). However, Helgeson et al. fails to disclose a method comprising the steps of: using an external configuration file to determine

whether a particular verb that is included in the automation message is a valid verb, wherein said configuration file includes a plurality of verbs and an association of each one of said plurality of verbs to one of said automation tools, and wherein said particular verb is valid if said particular verb is included in said configuration file; responsive to determining that said particular verb is a valid verb, said queue listener creating a verb dispatcher within said automation tool adapter; passing, by said queue listener, said automation message and said particular verb to said verb dispatcher; determining whether an idle execution thread is available from a thread pool that is assigned to one of a plurality of verb groups to which the particular verb belongs, and wherein each one of said plurality of verbs belongs to one of said plurality of verb groups, and further wherein said configuration file defines a maximum number of concurrent threads that are available to each one of the plurality of verb groups; responsive to determining that said idle execution thread is available, executing said verb dispatcher on said idle execution thread; responsive to determining that an idle execution thread is not available, determining whether said maximum number of concurrent threads available to said one of said plurality of verb groups has been reached; responsive to determining that said maximum number of concurrent threads available to said one of said plurality of verb groups has not been reached, creating a new thread and executing said verb dispatcher on said new thread. Patiejunas discloses a system and method for implementing a client side HTTP stack comprising the steps of: using an external configuration file to determine whether a particular verb that is included in the automation message is a valid verb, wherein said configuration file includes a plurality

of verbs and an association of each one of said plurality of verbs to one of said automation tools, and wherein said particular verb is valid if said particular verb is included in said configuration file ("The CCommunicationThreadPool thread pool 80 may be used publicly via an application program interface (API). The following table describes several exemplary methods which may be available for public use: 2

Method/data member	Description
CCommunicationThreadPool( )	Creates thread pool with default configuration
CCommunicationThreadPool( )	Creates custom configured thread pool (read as configuration file)
DWORD a_dwNofCompletionThreads, DWORD a_dwNofConnectionThreads, DWORD a_dwTimeoutPrecission, DWORD a_dwTimeoutPeriod	
.about.CCommunicationThreadPool( )	Destroys thread pool
HRESULT Initialize ( )	Performs initialization of thread pool. Only after this moment all data structures are allocated and threads started.
HRESULT ShutDown ( )	Shutowns thread pool by stopping threads and clearing allocations. Depending on refcount connections object will be freed
PMASSIVE_COMM_THREAD_STATS	Returns a pointer to the global engine statistics
QueryStatistics( )	structure VOID
ResetStatistics( )	Resets engine statistics")

Patiejunas, paragraph 0056); responsive to determining that said particular verb is a valid verb, said queue listener creating a verb dispatcher within said automation tool adapter ("Thus, many server side HTTP stack software implementations provide for multithreading, which allows two or more streams of execution (threads) to run concurrently within a single program (e.g., to service multiple client requests). Multithreading may be employed in a single processor server machine, and/or in a multiprocessing environment wherein a plurality of processors may service

individual threads. For example, individual client requests may be processed by corresponding threads, thereby increasing the request handling capacity of the server. In addition, some server side HTTP stack implementations include methods and software components for facilitating efficient usage of such threads, for example, completion ports and the like.") Patiejunas, paragraph 0005); passing, by said queue listener, said automation message and said particular verb to said verb dispatcher; determining whether an idle execution thread is available from a thread pool that is assigned to one of a plurality of verb groups to which the particular verb belongs, and wherein each one of said plurality of verbs belongs to one of said plurality of verb groups, and further wherein said configuration file defines a maximum number of concurrent threads that are available to each one of the plurality of verb groups; responsive to determining that said idle execution thread is available, executing said verb dispatcher on said idle execution thread; responsive to determining that an idle execution thread is not available, determining whether said maximum number of concurrent threads available to said one of said plurality of verb groups has been reached; responsive to determining that said maximum number of concurrent threads available to said one of said plurality of verb groups has not been reached, creating a new thread and executing said verb dispatcher on said new thread ("The invention allows client side applications to create completion ports with an associated concurrency value indicating the maximum number of threads associated with the port which should be running at any given time. I/O is associated with client side sockets, which in turn are associated with the completion port using a key. The use of completion



ports and concurrency values improves processor utilization by allowing blocking threads to be deactivated, thereby suspending execution of tasks related to a given request, until the completion port receives an associated completion packet from the I/O. In operation, the threads that block on a completion port are deactivated, thereby allowing other threads to be activated as completion packets are received at the completion port within the concurrency limits. In addition, the client side HTTP stack implementation provides for state machines associated with the requests. The state machines are associated with specific requests using one or more keys. When a client side completion port receives a completion packet from a server, the next available thread processes the request according to the corresponding state machine using the key. The state machine allows the correct processing of tasks associated with a particular request by any one of the threads from the thread pool, and thus facilitates the improved processing efficiencies achieved through the use of a thread pool and completion ports. In particular, the key facilitates the ability of a thread whose associated operation (e.g., an I/O operation) is pending, to check a completion port which may then activate the thread when any other operation is completed. When the initial (e.g., I/O) operation completes, the next available thread then resumes execution thereof at the appropriate state using the key. The thread thus returns to a pool of available or free threads once the thread receives an indication (e.g., status code) that the current operation is pending.”) Patiejunas, paragraphs 0010-0011 and 0067);

Helgeson et al. discloses a prior art method for a business application server management system platform providing a single automation tool adapter within said business application server management system platform for use with multiple different automation tools, said method comprising the steps of: each one of said automation tools being an application that interfaces with a device to enact a sequence of automated changes; said automation tool adapter including a queue listener; monitoring, by said queue listener, a plurality of clients by monitoring a request queue that is external to said automation tool adapter for messages; determining, by said queue listener, whether said request queue includes an automation message; said automation message including a request having parameters in an original format that are to be executed using one of said automation tools, said automation message including a message identifier that uniquely identifies said automation message and a source of said automation message; responsive to said queue listener determining that said request queue includes said automation message, said queue listener creating a verb dispatcher within said automation tool adapter; passing, by said queue listener, said automation message to said verb dispatcher; creating, by said verb dispatcher, an automation processor of a type that is required by said particular verb; translating, by said verb dispatcher, said parameters from said original format to a second format that is required by said one of said automation tools to produce translated parameters; providing, by said verb dispatcher, said message including said message identifier and said translated parameters to said automation processor; constructing, by said automation processor, an automation tool command using said translated parameters;

executing, by said one of said automation tools, said automation tool command, said automation tool command including said translated parameters; in response to a completion of execution of said automation tool command, said one of said automation tools sending a reply to a reply server that is included in said automation tool adapter, said reply being in said second format and including said message identifier; translating, by said reply server, said reply into said original format to form a translated reply; and sending, by said reply server, said translated reply to a reply queue that is external to said automation tool adapter upon which the claimed invention can be seen as an improvement.

Patiejunas teaches a prior art comparable method comprising the steps of: using an external configuration file to determine whether a particular verb that is included in the automation message is a valid verb, wherein said configuration file includes a plurality of verbs and an association of each one of said plurality of verbs to one of said automation tools, and wherein said particular verb is valid if said particular verb is included in said configuration file; responsive to determining that said particular verb is a valid verb, said queue listener creating a verb dispatcher within said automation tool adapter; passing, by said queue listener, said automation message and said particular verb to said verb dispatcher; determining whether an idle execution thread is available from a thread pool that is assigned to one of a plurality of verb groups to which the particular verb belongs, and wherein each one of said plurality of verbs belongs to one of said plurality of verb groups, and further wherein said configuration file defines a

maximum number of concurrent threads that are available to each one of the plurality of verb groups; responsive to determining that said idle execution thread is available, executing said verb dispatcher on said idle execution thread; responsive to determining that an idle execution thread is not available, determining whether said maximum number of concurrent threads available to said one of said plurality of verb groups has been reached; responsive to determining that said maximum number of concurrent threads available to said one of said plurality of verb groups has not been reached, creating a new thread and executing said verb dispatcher on said new thread.

Thus, the manner of enhancing a particular method comprising the steps of using an external configuration file to determine whether a particular verb that is included in the automation message is a valid verb, wherein said configuration file includes a plurality of verbs and an association of each one of said plurality of verbs to one of said automation tools, and wherein said particular verb is valid if said particular verb is included in said configuration file; responsive to determining that said particular verb is a valid verb, said queue listener creating a verb dispatcher within said automation tool adapter; passing, by said queue listener, said automation message and said particular verb to said verb dispatcher; determining whether an idle execution thread is available from a thread pool that is assigned to one of a plurality of verb groups to which the particular verb belongs, and wherein each one of said plurality of verbs belongs to one of said plurality of verb groups, and further wherein said configuration file defines a maximum number of concurrent threads that are available to each one of the plurality of

verb groups; responsive to determining that said idle execution thread is available, executing said verb dispatcher on said idle execution thread; responsive to determining that an idle execution thread is not available, determining whether said maximum number of concurrent threads available to said one of said plurality of verb groups has been reached; responsive to determining that said maximum number of concurrent threads available to said one of said plurality of verb groups has not been reached, creating a new thread and executing said verb dispatcher on said new thread) was made part of the ordinary capabilities of one skilled in the art based upon the teaching of such improvement in Patiejunas. Accordingly, one of ordinary skill in the art would have been capable of applying this known improvement technique in the same manner to the prior art method for a business application server management system platform providing a single automation tool adapter within said business application server management system platform for use with multiple different automation tools, said method comprising the steps of: each one of said automation tools being an application that interfaces with a device to enact a sequence of automated changes; said automation tool adapter including a queue listener; monitoring, by said queue listener, a plurality of clients by monitoring a request queue that is external to said automation tool adapter for messages; determining, by said queue listener, whether said request queue includes an automation message; said automation message including a request having parameters in an original format that are to be executed using one of said automation tools, said automation message including a message identifier that uniquely identifies said automation message and a source of said automation message; responsive to said

queue listener determining that said request queue includes said automation message, said queue listener creating a verb dispatcher within said automation tool adapter; passing, by said queue listener, said automation message to said verb dispatcher; creating, by said verb dispatcher, an automation processor of a type that is required by said particular verb; translating, by said verb dispatcher, said parameters from said original format to a second format that is required by said one of said automation tools to produce translated parameters; providing, by said verb dispatcher, said message including said message identifier and said translated parameters to said automation processor; constructing, by said automation processor, an automation tool command using said translated parameters; executing, by said one of said automation tools, said automation tool command, said automation tool command including said translated parameters; in response to a completion of execution of said automation tool command, said one of said automation tools sending a reply to a reply server that is included in said automation tool adapter, said reply being in said second format and including said message identifier; translating, by said reply server, said reply into said original format to form a translated reply; and sending, by said reply server, said translated reply to a reply queue that is external to said automation tool adapter of Helgeson et al. and the results would have been predictable to one of ordinary skill in the art, namely, one skilled in the art would have readily recognized an event generator.

Consider claim 4, as applied to claims 1. Helgeson et al. discloses a method comprising the steps of: determining said second format required by said one of said

automation tools utilizing a configuration file that includes a specification of a format for each one of said automation tools (Helgeson et al., column 59 line 37 – column 60 line 53 and column 61 lines 27-43).

Consider claims 5, as applied to claims 1. Helgeson et al. discloses a method comprising the steps of: said original format being an XML document format; and translating said parameters from said XML document format to a flat file containing a set of name/value pairs, wherein said flat file format is said second format (Helgeson et al., column 28 line 48 – column 29 line 55).

**8.** Claims 2-3 are rejected under 35 U.S.C. 103(a) as being unpatentable over Helgeson et al. (US 7072934 B2) in view of Patiejunas (US 20050108710 A1) and in further view of Belfiore et al. (US 6990513 B2).

Consider claims 2, as applied to claims 1. Helgeson et al., as modified by Patiejunas, discloses a method for a business applications server management system platform comprising request queues. However, Helgeson et al., as modified by Patiejunas, fails to disclose a plurality of request queues. Belfiore et al. discloses a distributed computing services platform comprising: providing a plurality of request queues; said automation tool adapter including a plurality of queue listeners, each one of said plurality of request queues associated with one of said plurality of queue listeners; and checking, by each one of said plurality of queue listeners, an associated

one of said plurality of request queues for a message (Belfiore et al., column 20 lines 45-56).

Therefore, it would have been obvious for a person of ordinary skill in the art at the time the invention was made to incorporate a distributed computing services platform comprising: providing a plurality of request queues; said automation tool adapter including a plurality of queue listeners, each one of said plurality of request queues associated with one of said plurality of queue listeners; and checking, by each one of said plurality of queue listeners, an associated one of said plurality of request queues for a message as taught by Belfiore et al. with a method for a business applications server management system platform comprising request queues as taught by Helgeson et al., as modified by Patiejunas, for the purpose of distributed applications computing.

Consider claim 3, as applied to claims 2. Helgeson et al., as modified by Belfiore et al. and Patiejunas, discloses a method comprising the steps of: each one of said plurality of request queues located in a different one of said plurality of clients (Helgeson et al., column 31 line 60 – column 32 line 12).

### ***Response to Arguments***

9. Applicant's arguments filed 13 August 2008 with respect to claims 1-5 have been considered but are moot in view of the new ground(s) of rejection.



***Conclusion***

**10.** Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the mailing date of this final action.

Any response to this Office Action should be faxed to (571) 273-8300 or mailed to:

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Hand-delivered responses should be brought to

Customer Service Window  
Randolph Building  
401 Dulany Street  
Alexandria, VA 22314

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Mark Fearer whose telephone number is (571) 270-1770. The Examiner can normally be reached on Monday-Thursday from 7:30am to 5:00pm.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Tonia Dollinger can be reached on (571) 272-4170. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free) or 571-272-4100.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist/customer service whose telephone number is (571) 272-2600.

Mark Fearer  
/M.D.F./  
November 5, 2008

Application/Control Number: 10/671,215

Page 18

Art Unit: 2443

/Tonia LM Dollinger/

Supervisory Patent Examiner, Art Unit 2443